Deep Learning

Yann LeCun, Yoshua Bengio & Geoffrey Hinton

Nature 521, 436–444 (28 May 2015)
doi:10.1038/nature14539

Authors' Relationships

Geoffrey Hinton
1947, Google & U of T, BP
92.9-93.10 >200 papers

Michael I. Jordan
1956, UC Berkeley

PhD

Andrew NG(吴恩达)
1976, Stanford, Coursera
Google Brain → Baidu Brain

PhD

postdoc

AT&T
colleague

Yann LeCun
1960, Facebook & NYU,
CNN & LeNet

Yoshua Bengio
1964, UdeM, RNN & NLP

PhD

postdoc

postdoc

PhD

Abstraction

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction.

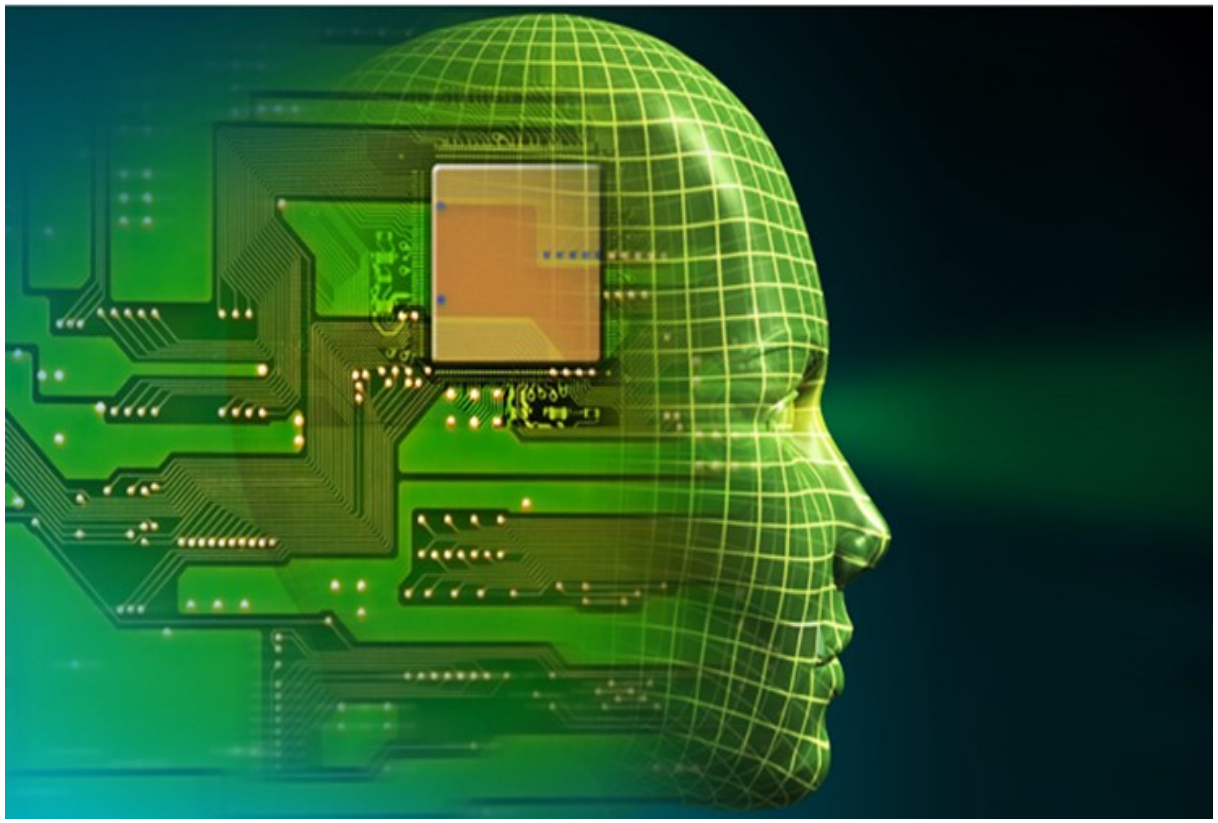
These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics.

Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer.

Deep convolutional nets have brought about breakthroughs in processing images, video, speech and audio, whereas recurrent nets have shone light on sequential data such as text and speech.

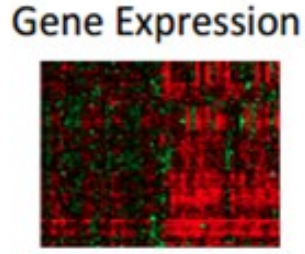
Deep Learning

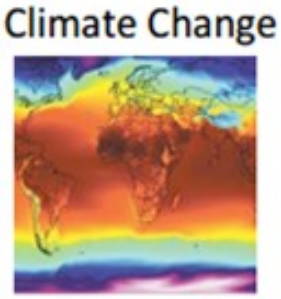
- Definition
- Applied Domains
  - Speech recognition, …
- Mechanism
- Networks
  - Deep Convolutional Nets (CNN)
  - Deep Recurrent Nets (RNN)

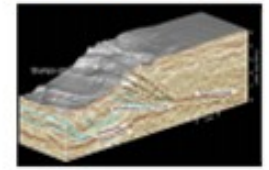


Applied Domains

- Speech Recognition
  - Speech → Words
- Visual Object Recognition
  - ImageNet (car, dog)
- Object Detection
  - Face detection
  - pedestrian detection
- Drug Discovery
  - Predict drug activity
- Genomics
  - Deep Genomics company







Conventional Machine Learning

- Limited in their ability to process natural data in their raw form.
- Feature!!!
  - Coming up with features is difficult, time-consuming, requires expert knowledge.
  - When working applications of learning, we spend a lot of time tuning the features.

Representation Learning

- Representation learning
  - A machine be fed with raw data
  - Automatically discover representations
- Deep-learning methods are representation-learning methods with multiple levels of representation
  - Simple but non-linear modules → higher and abstract representation
  - With the composition of enough such transformations, very complex functions can be learned.
- Key aspect
  - Layers of features are not designed by human engineers.
  - Learn features from data using a general-purpose learning procedure.

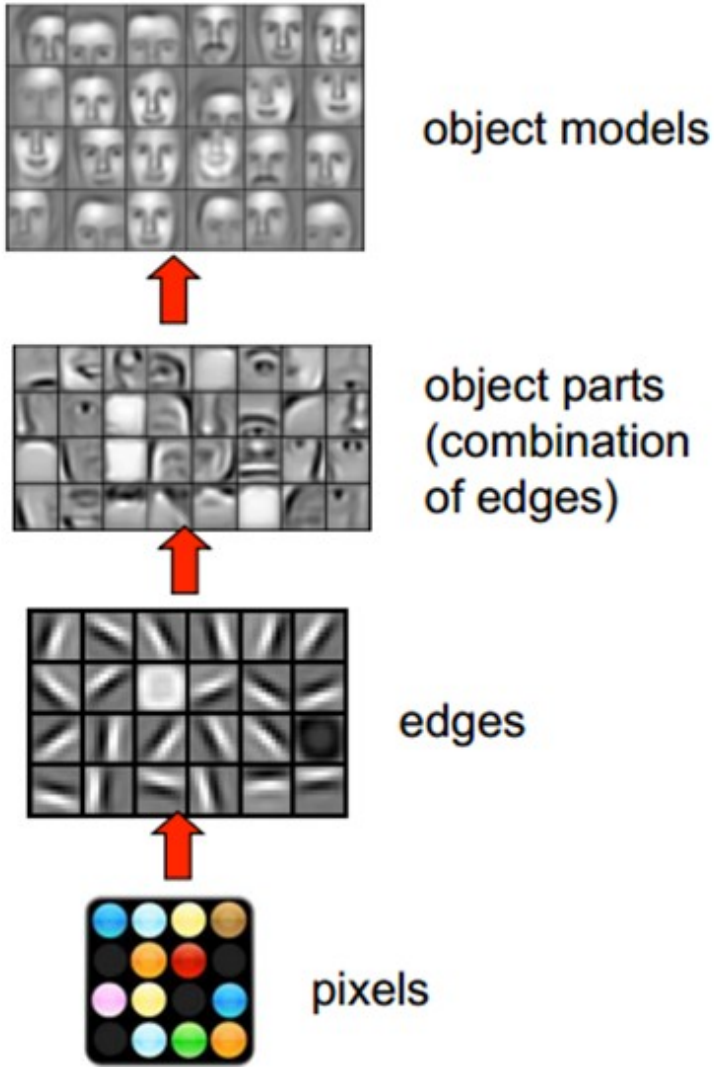
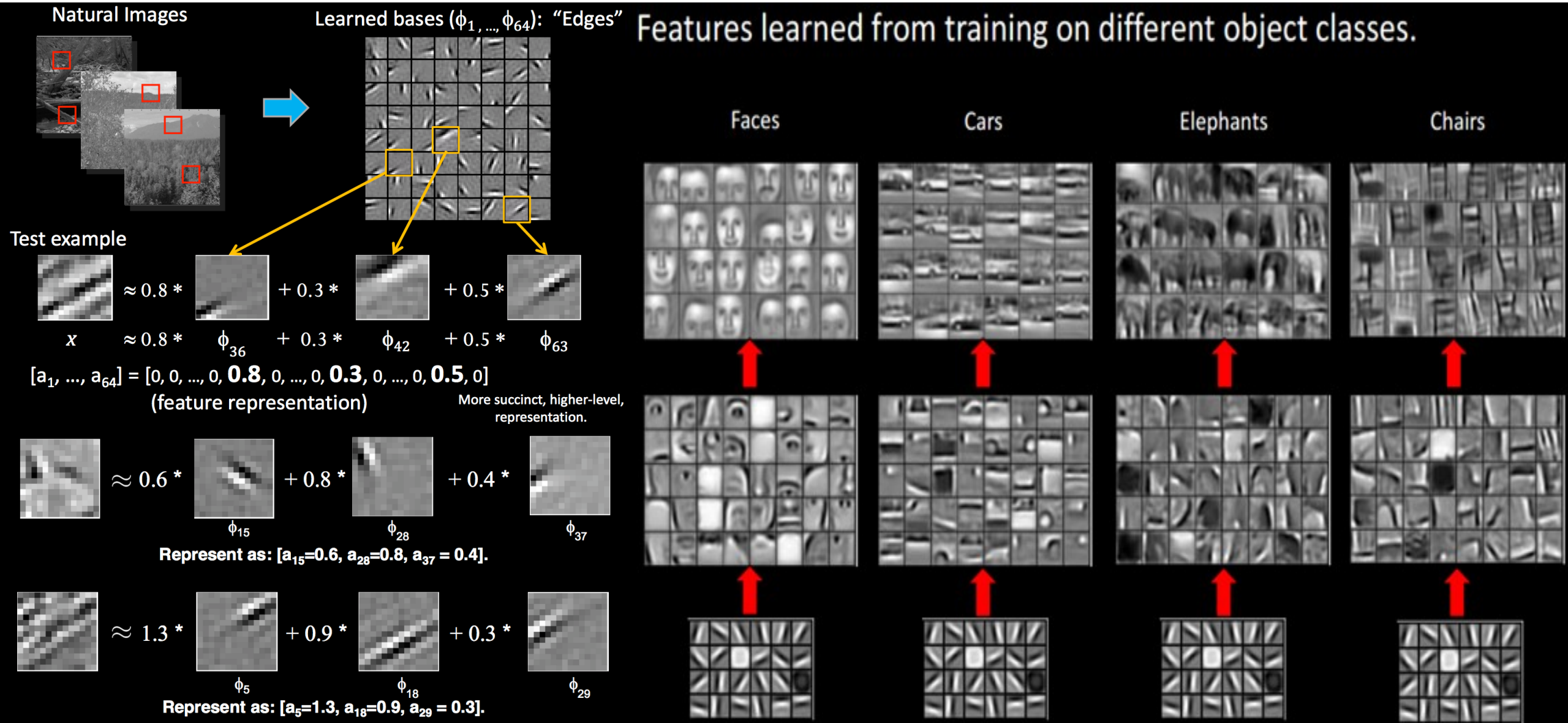
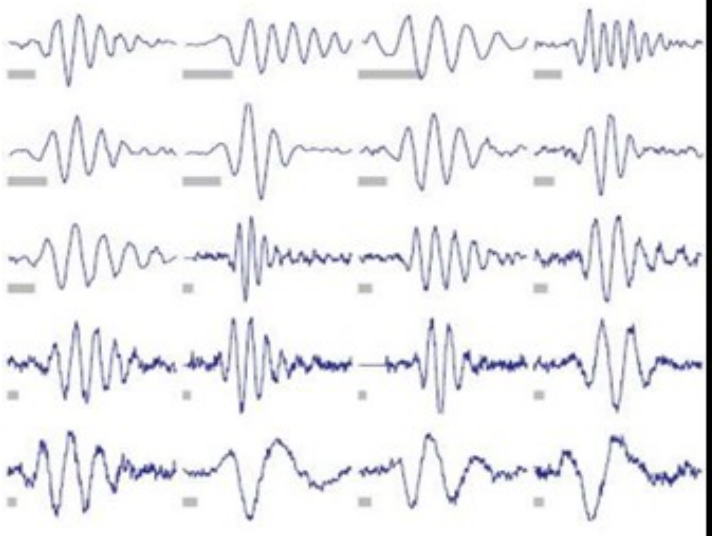


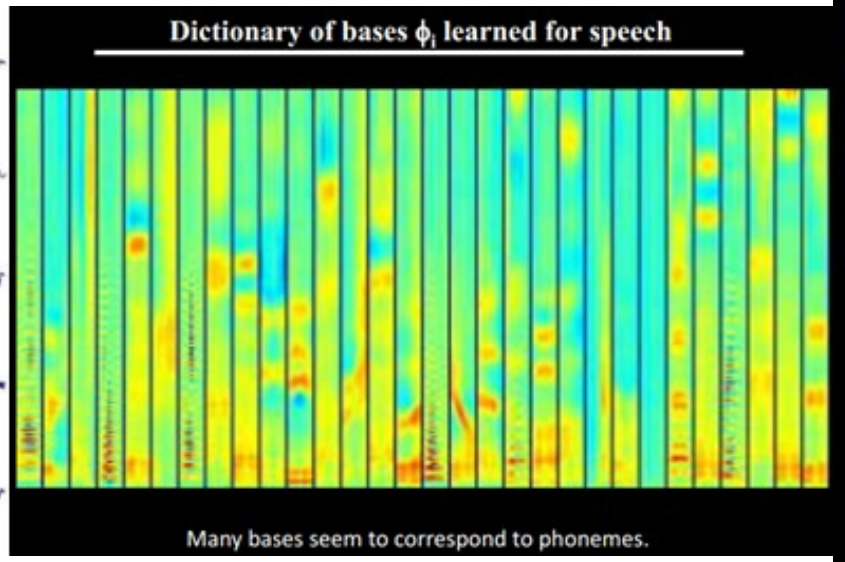
Image Features

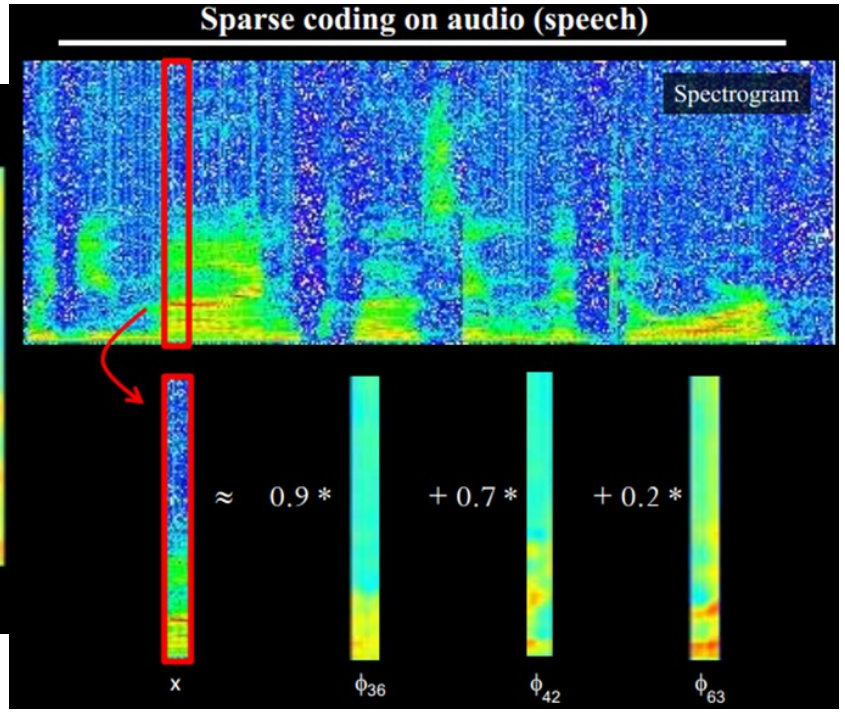


Audio Features

- 20 basic audio structure







Advances

- Image recognition
  - Hinton, 2012, ref. 1; LeCun, 2013, ref. 2; LeCun, 2014, ref. 3; Szegedy, 2014, ref. 4
- Speech recognition
  - Mikolov, 2011, ref. 5; Hinton, 2012, ref. 6; Sainath, 2013, ref. 7
- Many domains
  - Predicting the activity of potential drug molecules. Ma, J., 2015, ref. 8
  - Analysing particle accelerator data. Ciodaro, 2012, ref. 9; Kaggle, 2014, ref. 10
  - Reconstructing brain circuits. Helmstaedter, 2013, ref. 11
  - Predicting the effects of mutations in non-coding DNA on gene expression and disease. Leung, 2014, ref. 12; Xiong, 2015, ref. 13
- Natural language understanding(Collobert, 2011, ref. 14)
  - Topic classification.
  - Sentiment analysis.
  - Question answering. Bordes, 2014, ref. 15
  - Language translation. Jean, 2015, ref. 16; Sutskever, 2014, ref. 17

Outline

- Supervised learning
- Backpropagation to train multilayer architectures
- Convolutional neural networks
- Image understanding with deep convolutional networks
- Distributed representations and language processing
- Recurrent neural networks
- The future of deep learning

Supervised Learning

- Procedures
  - dataset → labeling → training (errors, tuning parameters, gradient descent) → testing
- Stochastic gradient descent (SGD, Bottou, 2007, ref. 18)
  - Showing input vector, computing outputs and errors, computing average gradient, adjusting weights accordingly
  - Repeating for many small sets until objective function stop decreasing
  - Why stochastic: small set gives a noisy estimate of the average gradient over all examples
- Linear classifiers or shallow classifiers (must have good features)
  - input space → half-spaces (Duda, 1973, ref. 19): cannot distinguish wolf and Samoyed in same position and background
  - kernel methods (Scholkopf, 2002, ref. 20; Bengio, 2005, ref. 21): do not generalize well
- Deep learning architecture
  - multiple non-linear layers (5-20): can distinguish Samoyeds from white wolves

Backpropagation to Train Multilayer Architectures

- Key insight: (ref. 22-27)
  - Nothing more than a practical application of the chain rule for derivatives.
- Feedforward neural networks
  - Non-linear function: max(z, 0) (ReLU, Begio, 2011, ref. 28), tanh(z), 1/(1+exp(-z))
- Forsaken because poor local minima
- Revived around 2006 by unsupervised learning procedures with unlabeled data
  - In practice, local minima are rarely a problem. (Dauphin, 2014, ref. 29; LeCun, 2014, ref. 30)
  - CIFAR match: Hinton, 2005 ref. 31; Hinton, 2006, ref. 32; Bengio, 2006, ref. 33; LeCun, 2006, ref. 34; Hinton, 2006, ref. 35
  - Recognizing handwritten digits or detecting pedestrians (LeCun, 2013, ref. 36)
  - Speech recognition by GPUs with 10 or 20 times faster (Raina, 2009, ref. 37; Hinton, 2012, ref. 38; Dahl, 2012, ref. 39; Bengio, 2013, ref. 40)
- Convolutional neural network (ConvNet, CNN)
  - Widely adopted by computer-vision community: LeCun, 1990, ref. 41; LeCun, 1998, ref. 42

BP Key Insight

- The derivative (or gradient) of the objective with respect to the input of a module can be computed by working backwards from the gradient with respect to the output of that module (or the input of the subsequent module).

b

$$\Delta z = \frac{\partial z}{\partial y}\Delta y$$

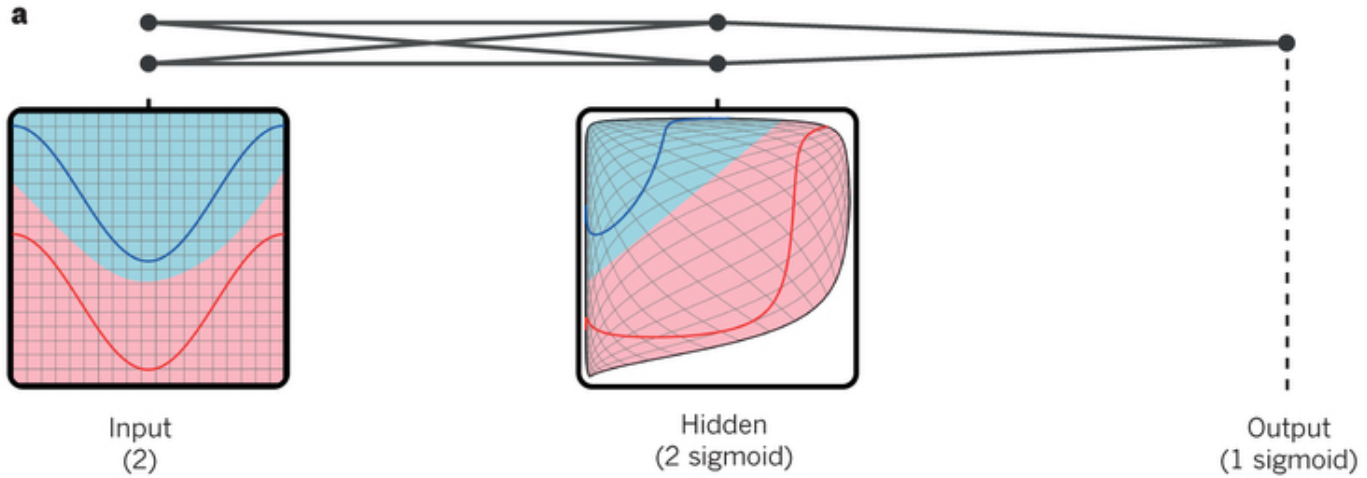
$$\Delta y = \frac{\partial y}{\partial x}\Delta x$$

$$\Delta z = \frac{\partial z}{\partial y}\frac{\partial y}{\partial x}\Delta x$$

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y}\frac{\partial y}{\partial x}$$

b. The chain rule of derivatives tells us how two small effects (that of a small change of x on y, and that of y on z) are composed. A small change Δx in x gets transformed first into a small change Δy in y by getting multiplied by $\partial y/\partial x$ (that is, the definition of partial derivative). Similarly, the change Δy creates a change Δz in z. Substituting one equation into the other gives the chain rule of derivatives — how Δx gets turned into Δz through multiplication by the product of $\partial y/\partial x$ and $\partial z/\partial x$. It also works when x, y and z are vectors (and the derivatives are Jacobian matrices).

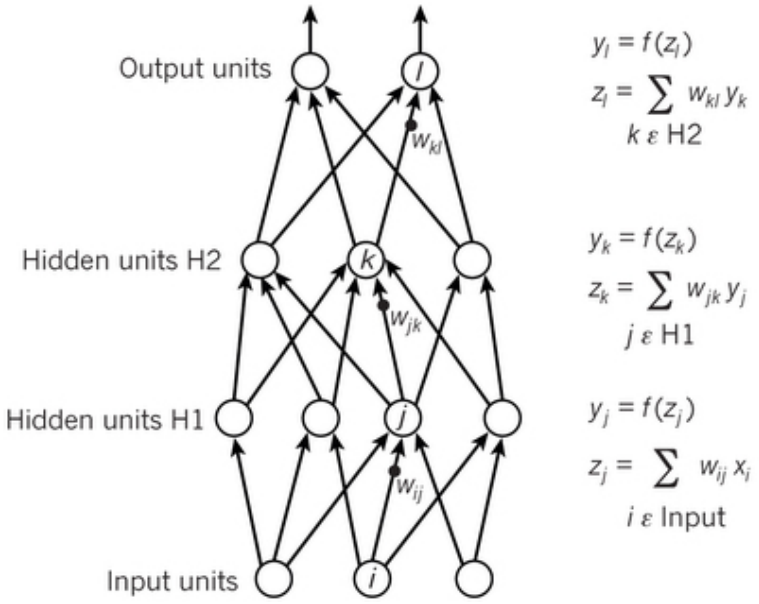
Multilayer neural network



a. A multilayer neural network (shown by the connected dots) can distort the input space to make the classes of data (examples of which are on the red and blue lines) linearly separable. Note how a regular grid (shown on the left) in input space is also transformed (shown in the middle panel) by hidden units. This is an illustrative example with only two input units, two hidden units and one output unit, but the networks used for object recognition or natural language processing contain tens or hundreds of thousands of units. Reproduced with permission from C. Olah (http://colah.github.io/).

Feedforward

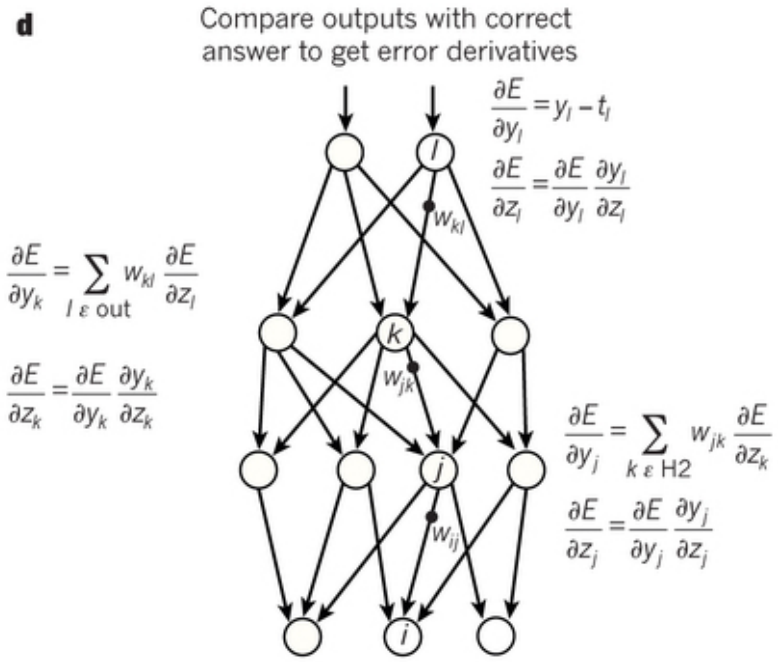
c



c. The equations used for computing the forward pass in a neural net with two hidden layers and one output layer, each constituting a module through which one can backpropagate gradients. At each layer, we first compute the total input *z* to each unit, which is a weighted sum of the outputs of the units in the layer below. Then a non-linear function *f(.)* is applied to z to get the output of the unit. For simplicity, we have omitted bias terms. The non-linear functions used in neural networks include the rectified linear unit (ReLU), commonly used in recent years, as well as the more conventional sigmoids, such as the hyberbolic tangent (tanh), logistic function.

- ReLU: $f(z) = \max(z, 0)$
- Hyberbolic tangent: $f(z) = \frac{\exp(\mathrm{x}) - \exp(-z)}{\exp(z) + \exp(-z)}$
- logistic function: $f(z) = \frac{1}{1 + \exp(-z)}$

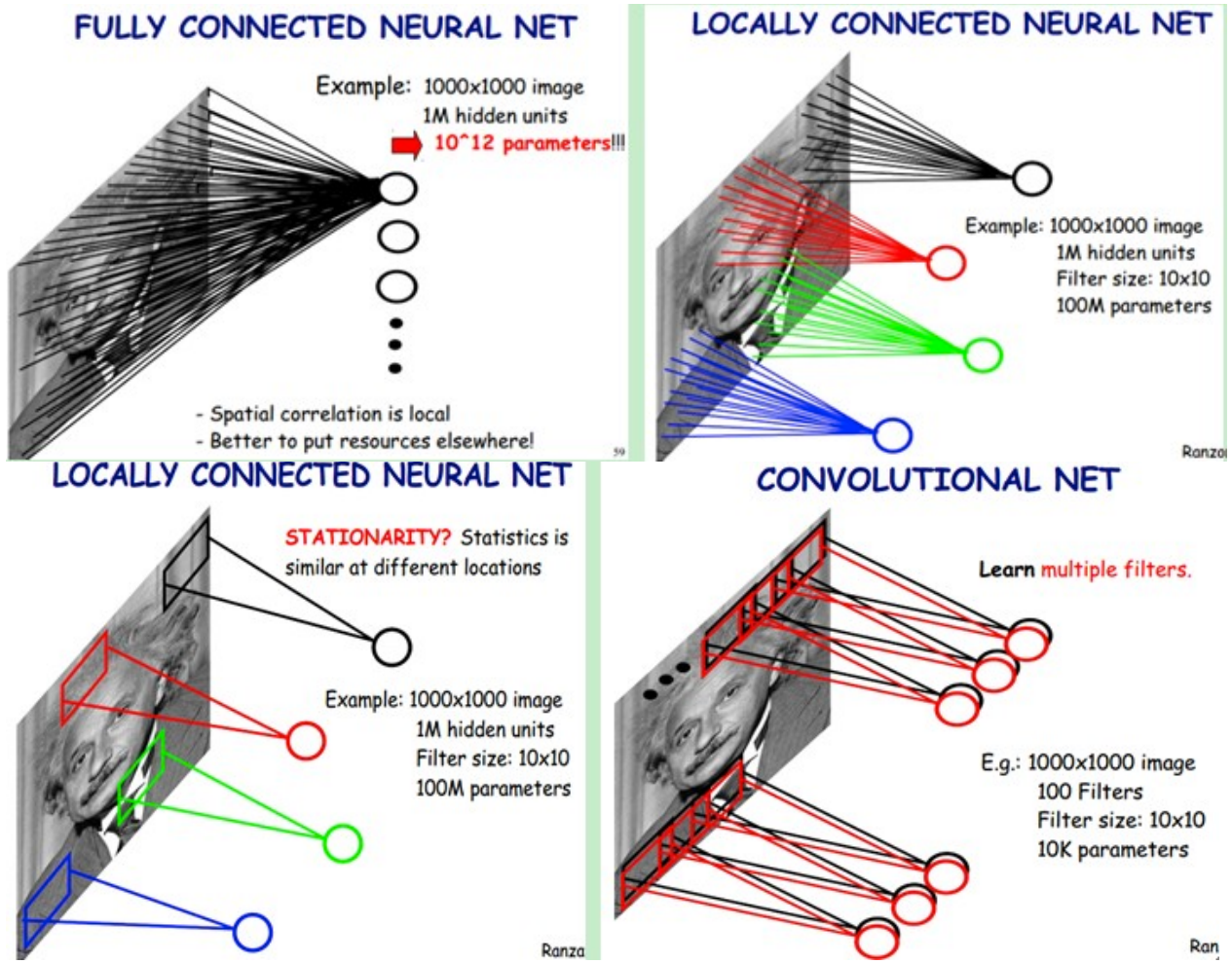
Backpropagation



d. The equations used for computing the backward pass. At each hidden layer we compute the error derivative with respect to the output of each unit, which is a weighted sum of the error derivatives with respect to the total inputs to the units in the layer above. We then convert the error derivative with respect to the output into the error derivative with respect to the input by multiplying it by the gradient of $f(z)$. At the output layer, the error derivative with respect to the output of a unit is computed by differentiating the cost function. This gives $y_l - t_l$ if the cost function for unit l is $0.5(y_l - t_l)^2$, where t_l is the target value. Once the $\partial E/\partial z_k$ is known, the error-derivative for the weight w_{jk} on the connection from unit j in the layer below is just $y_j \, \partial E/\partial z_k$.

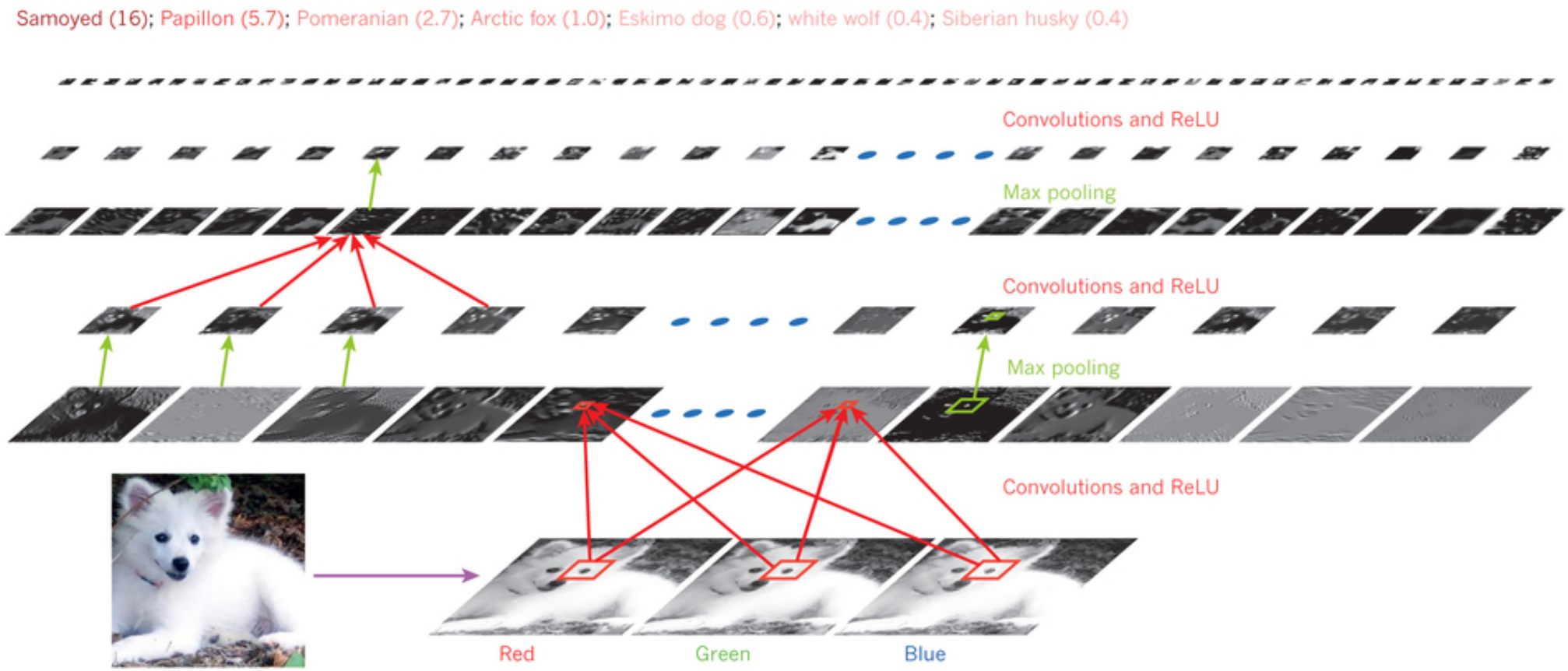
Convolutional Neural Networks

- Local connections
  - local values are correlated
- Shared weights
  - local statistics are invariant to location
- Pooling
  - merge similar features into one
- The use of many layers
  - many layers of convolutional, non-linearity and pooling



Mathematically, the filtering operation performed by a feature map is a discrete convolution, hence the name.

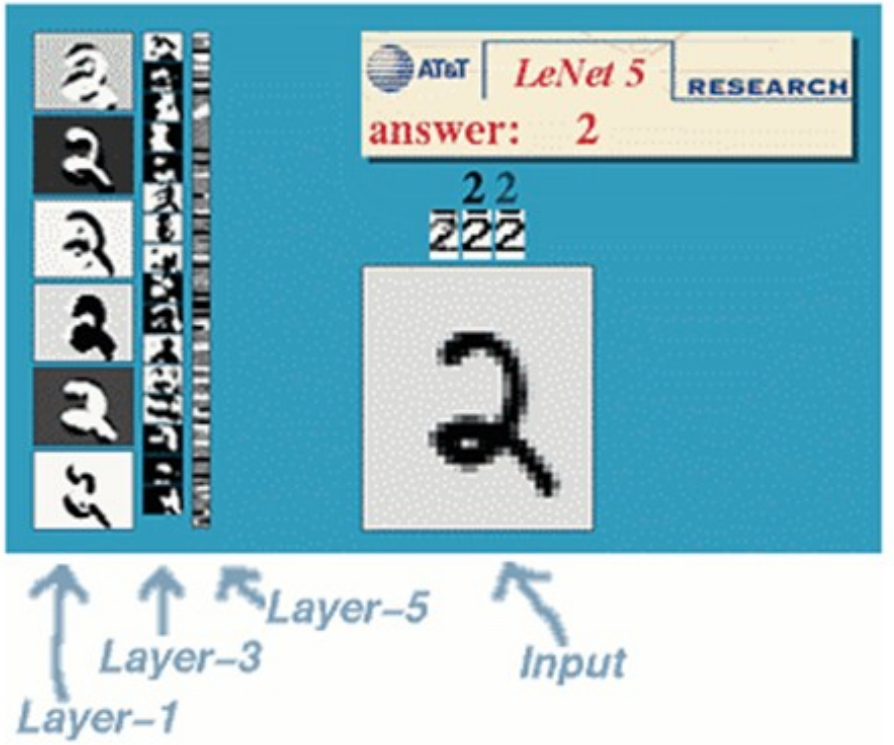
Inside A Convolutional Network



The outputs (not the filters) of each layer (horizontally) of a typical convolutional network architecture applied to the image of a Samoyed dog (bottom left; and RGB (red, green, blue) inputs, bottom right). Each rectangular image is a feature map corresponding to the output for one of the learned features, detected at each of the image positions. Information flows bottom up, with lower-level features acting as oriented edge detectors, and a score is computed for each image class in output. ReLU, rectified linear unit.

LeNet-5

- LeNet-5 (LeCun, 1998)
  - C1: 5*5 units → 1 unit; 32*32 → 28*28; 5*5*6*1+*6=156 parameters; (5*5+1)*6*(28*28)=122,304 connections; 1 map → 6 maps
  - S2: 2*2 units → 1 unit; 28*28 → 14*14 (no overlap); 2*6=12 parameters; (2*2+1)*6*(14*14)=5,880 connections; 1 map → 1 map
  - C3: 6 or some maps → 1map
  - …
  - http://yann.lecun.com/exdb/lenet/



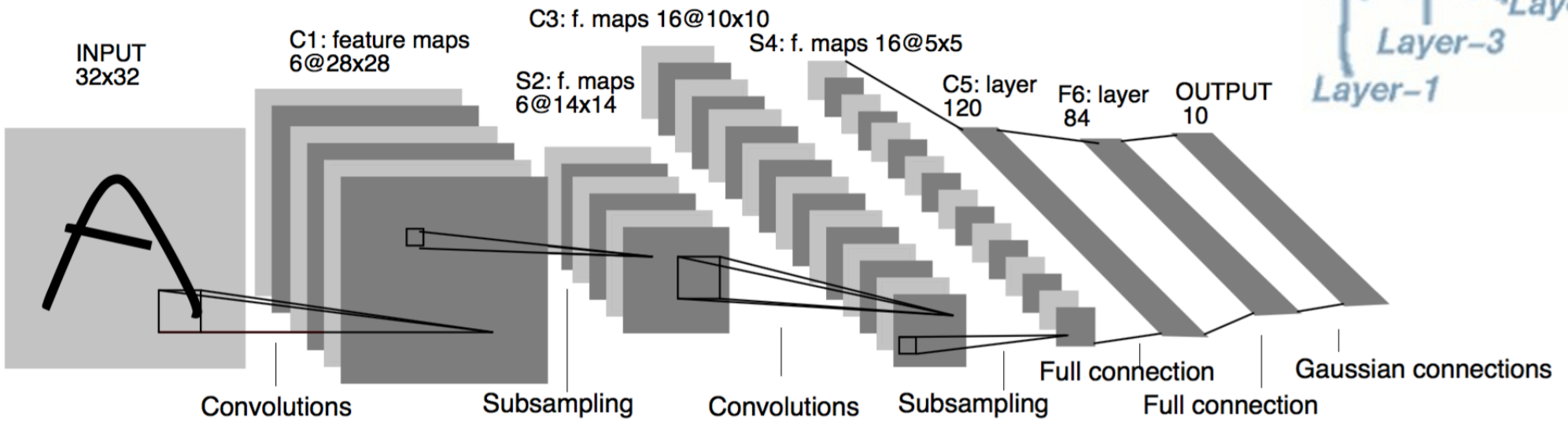
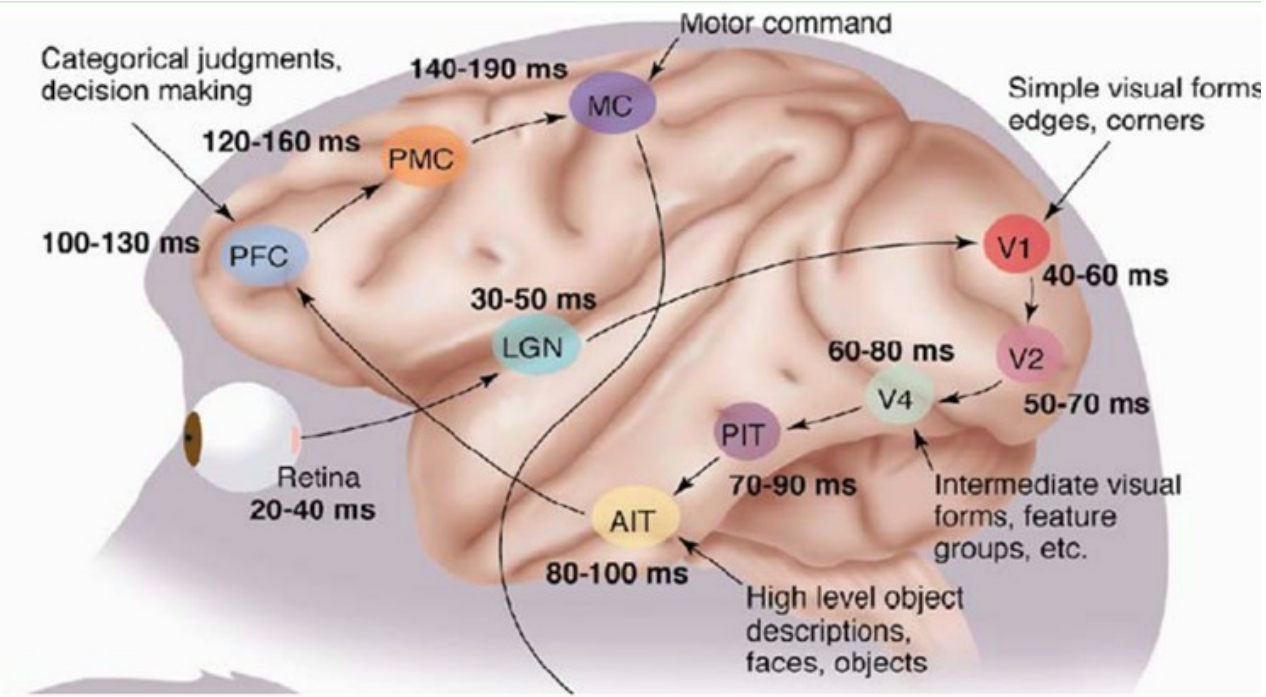
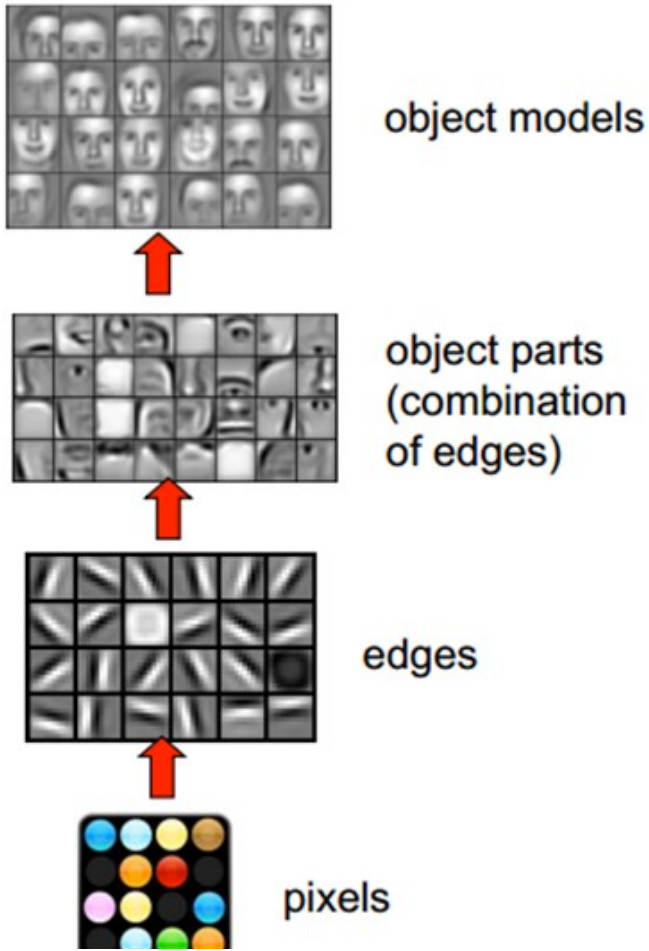


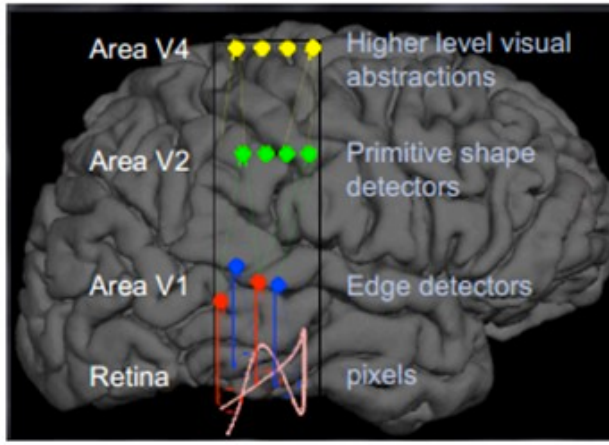
Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

ConvNets VS Visual Neuroscience

- Lower-level → higher-level
- LGN-V1-V2-V4-IT ventral pathway (Hubel, 1962, ref. 43; Felleman, 1991, ref. 44)
- Time-delay neural networks (ref. 45-48)
- Document reading, object detection, … (ref. 49-52)







DeepFace Architecture

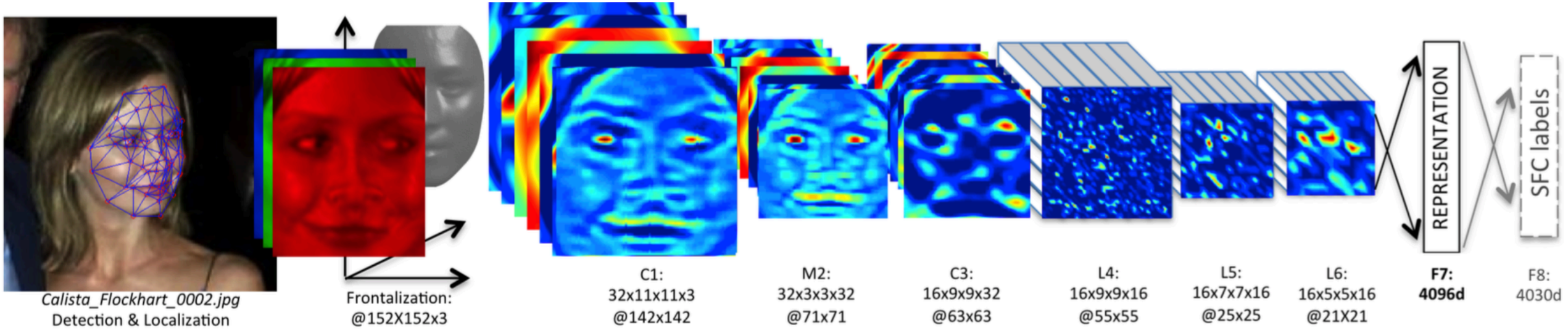


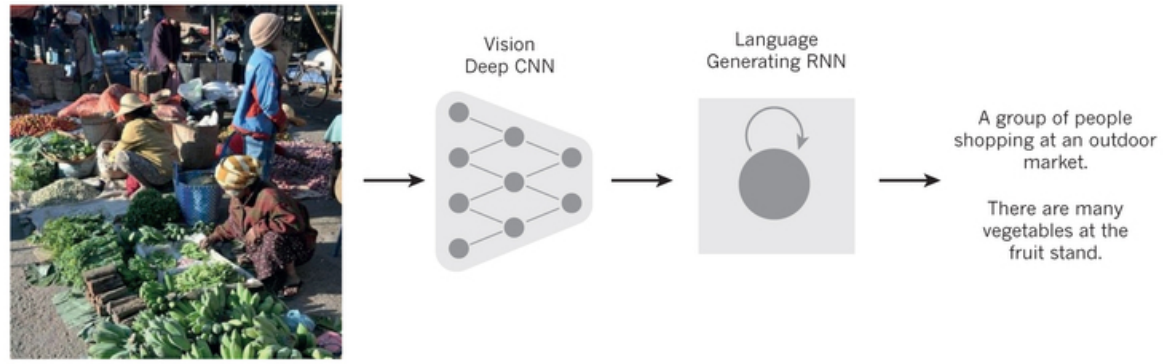
Figure 2. **Outline of the *DeepFace* architecture**. A front-end of a single convolution-pooling-convolution filtering on the rectified input, followed by three locally-connected layers and two fully-connected layers. Colors illustrate feature maps produced at each layer. The net includes more than 120 million parameters, where more than 95% come from the local and fully connected layers.

Methods	LFW 10-fold average precision	networks	datasets
DeepFace [Taigman, CVPR2014]	97.35%	3	4,000,000
DeepID [Sun, CVPR2014]	97.35%	25	200,000
DeepID2 [Sun, NIPS2014]	99.15%	25	200,000
DeepID2+ [Sun, CVPR2015]	99.47%	25	290,000
WSTFusion [Taigman, CVPR2015]	98.37%	-	1,000,000
VGGFace [Parkhi, BMVC2015]	98.95%	1	2,600,000
FaceNet [Schroff, CVPR 2015]	99.67%	1	200,000,000

Image Understanding with Deep Convolutional Networks

- Application
  - Traffic sign recognition, detect of pedestrians, … (ref. 36, ref. 50-51, ref. 53-58)
  - Face recognition (Deepface, Facebook, Taigman, CVPR, 2014, ref. 59)
  - Autonomous mobile robots and self-driving cars (Hadsell, 2009, ref. 60; Farabet, 2012, ref. 61)
  - Natural language understanding (Collobert, 2011, ref. 14)
  - Speech recognition (sainath, 2013, ref. 7)
- ConvNets in computer version
  - ImageNet (Hinton, 2012, ref. 1; Srivastava, 2014, ref. 62)
  - Other detection tasks (ref. 4, ref. 58-59, ref. 63-65)
- ConvNets-based product and service
  - Google, Facebook, Microsoft, IBM, Yahoo!, Twitter and Adobe
- ConvNets chips
  - NVIDIA, Mobileye, Intel, Qualcomm and Samsung

From Image to Text



Captions generated by a recurrent neural network (RNN) taking, as extra input, the representation extracted by a deep convolution neural network (CNN) from a test image, with the RNN trained to 'translate' high-level representations of images into captions (top). Reproduced with permission from ref. 102. When the RNN is given the ability to focus its attention on a different location in the input image (middle and bottom; the lighter patches were given more attention) as it generates each word (bold), we found that it exploits this to achieve better 'translation' of images into captions.

Distributed Representations and Language Processing

- Advantages
  - Learning distributed representations enable generalization to new combinations of the values of learned features beyond those seen during training (for example, 2^n combinations are possible with n binary features). (Bengio, 2009, ref. 68; Montufar, 2014, ref. 69)
  - Composing layers of representation in a deep net brings the potential for another exponential advantage (exponential in the depth). (Montufar, 2014, ref. 70)
- Applications
  - Neural language models predict the next word in a sequence (Begnio, 2001, ref. 71)
  - Such representations are called distributed representations because their elements (the features) are not mutually exclusive and their many configurations correspond to the variations seen in the observed data.
  - ref. 14, ref. 17, ref. 72-76

Visualizing The Learned Words Vectors

On the left is an illustration of word representations learned for modelling language, non-linearly projected to 2D for visualization using the t-SNE algorithm (ref. 103). On the right is a 2D representation of phrases learned by an English-to-French encoder–decoder recurrent neural network (ref. 75). One can observe that semantically similar words or sequences of words are mapped to nearby representations. The distributed representations of words are obtained by using backpropagation to jointly learn a representation for each word and a function that predicts a target quantity such as the next word in a sequence (for language modelling) or a whole sequence of translated words (for machine translation) (ref .18, ref. 75).

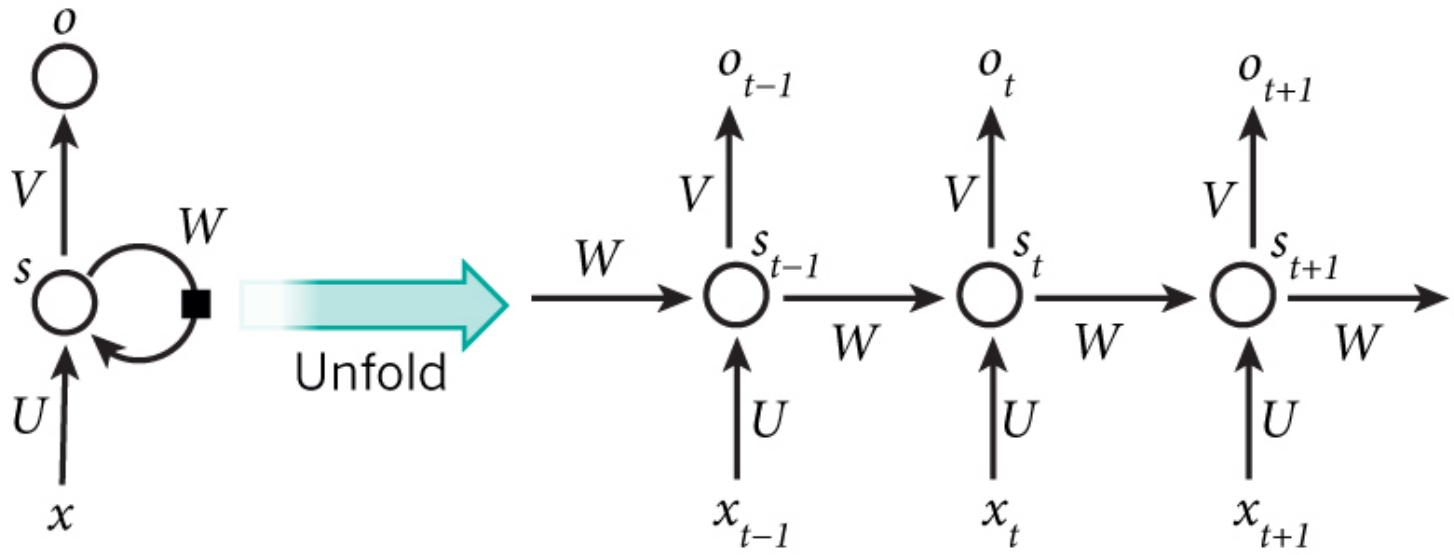
Recurrent Neural Networks

- It is better to use RNNs for tasks that involve sequential inputs, such as speech and language.
- Trained by backpropagation (ref. 81-82)
- Backpropagation gradients typically explode or vanish (ref. 77-78)
- Application:
  - Predict next character in the text (Stutskever, 2012, ref. 83)
  - Predict next word in a sequence (Lakoff, 2008, ref. 84)
  - English-French encoder-decoder network (ref. 17, ref. 72, ref. 76, ref.84-85)
- Very deep feedforward networks: difficult to learn story information very long. (ref. 78)
- Long short-term memory (LSTM) networks: memory cell (Hochreither, 1997, ref. 79)
  - More effective than conventional RNNs (Hinton, ref. 87)
- Neural Turning Machine (Graves, 2014, ref. 88) and Memory networks (Weston, 2014, ref. 89)
  - Memory networks can answer questions that require complex inference (Weston, 2015, ref. 90)

Memory Networks Answer questions

- 15-sentence version of *The Lord of the Rings*
  - Bilbo travelled to the cave. Gollum dropped the ring there. Bilbo took the ring. Bilbo went back to the Shire. Bilbo left the ring there. Frodo got the ring. Frodo journeyed to Mount-Doom. Frodo dropped the ring there. Sauron died. Frodo went back to the Shire. Bilbo travelled to the Grey-havens. The End.
- Question1:Where is the ring?
  - Answer: Mount-Doom
- Question2: Where is Bilbo now?
  - Answer: Grey-havens
- Question3: Where is Frodo now?
  - Answer: Shire

Recurrent Neural Network (RNN) and Unfolding



A recurrent neural network and the unfolding in time of the computation involved in its forward computation. The artificial neurons (for example, hidden units grouped under node s with values s_t at time t) get inputs from other neurons at previous time steps (this is represented with the black square, representing a delay of one time step, on the left). In this way, a recurrent neural network can map an input sequence with elements x_t into an output sequence with elements o_t, with each o_t depending on all the previous x_t' (for $t' \leq t$). The same parameters (matrices U,V,W) are used at each time step. Many other architectures are possible, including a variant in which the network can generate a sequence of outputs (for example, words), each of which is used as inputs for the next time step. The backpropagation algorithm can be directly applied to the computational graph of the unfolded network on the right, to compute the derivative of a total error (for example, the log-probability of generating the right sequence of outputs) with respect to all the states s_t and all the parameters.

The Future of Deep Learning

- Unsupervised learning
  - Had a catalytic effect in reviving interest in deep learning, but has since been overshadowed by the successes of purely supervised learning. (ref. 91-98)
  - Human and animal learning is largely unsupervised, we expect unsupervised learning to become far more important in the longer term.
- Computer vision
  - Combine ConvNets and RNNs and use reinforcement learning (ref. 99-100)
- Natural language understanding
  - RNNs: when they learn strategies for selectively attending to one part at a time. (ref. 76, ref. 86)
- Representation learning with complex reasoning
  - New paradigms are needed to replace rule-based manipulation of symbolic expressions by operations on large vectors. (ref. 101)

Important Reference

- Krizhevsky, A., Sutskever, I. & Hinton, G. ImageNet classification with deep convolutional neural networks. In Proc. Advances in Neural Information Processing Systems 25 1090–1098 (2012).
  - This report was a breakthrough that used convolutional nets to almost halve the error rate for object recognition, and precipitated the rapid adoption of deep learning by the computer vision community.
- Hinton, G. et al. Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Processing Magazine 29, 82–97 (2012).
  - This joint paper from the major speech recognition laboratories, summarizing the breakthrough achieved with deep learning on the task of phonetic classification for automatic speech recognition, was the first major industrial application of deep learning.
- Sutskever, I. Vinyals, O. & Le. Q. V. Sequence to sequence learning with neural networks. In Proc. Advances in Neural Information Processing Systems 27 3104–3112 (2014).
  - This paper showed state-of-the-art machine translation results with the architecture introduced in ref. 72, with a recurrent network trained to read a sentence in one language, produce a semantic representation of its meaning, and generate a translation in another language.

Important Reference II

- Glorot, X., Bordes, A. & Bengio. Y. Deep sparse rectifier neural networks. In Proc. 14th International Conference on Artificial Intelligence and Statistics 315–323 (2011).
  - This paper showed that supervised training of very deep neural networks is much faster if the hidden layers are composed of ReLU.
- Hinton,G.E.,Osindero,S.&Teh,Y.-W.Afastlearningalgorithmfordeepbelief nets. Neural Comp. 18, 1527–1554 (2006).
  - This paper introduced a novel and effective way of training very deep neural networks by pre-training one hidden layer at a time using the unsupervised learning procedure for restricted Boltzmann machines.
- Bengio,Y.,Lamblin,P.,Popovici,D.&Larochelle,H.Greedylayer-wisetraining of deep networks. In Proc. Advances in Neural Information Processing Systems 19 153–160 (2006).
  - This report demonstrated that the unsupervised pre-training method introduced in ref. 32 significantly improves performance on test data and generalizes the method to other unsupervised representation-learning techniques, such as auto-encoders.

Important Reference III

- LeCun,Y.etal. Handwrittendigitrecognitionwithaback-propagationnetwork. In Proc. Advances in Neural Information Processing Systems 396–404 (1990).
  - This is the first paper on convolutional networks trained by backpropagation for the task of classifying low-resolution images of handwritten digits.
- LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. Proc. IEEE 86, 2278–2324 (1998).
  - This overview paper on the principles of end-to-end training of modular systems such as deep neural networks using gradient-based optimization showed how neural networks (and in particular convolutional nets) can be combined with search or inference mechanisms to model complex outputs that are interdependent, such as sequences of characters associated with the content of a document.
- Bengio,Y.,Ducharme,R.&Vincent,P.Aneuralprobabilisticlanguagemodel.In Proc. Advances in Neural Information Processing Systems 13 932–938 (2001).
  - This paper introduced neural language models, which learn to convert a word symbol into a word vector or word embedding composed of learned semantic features in order to predict the next word in a sequence.

Important Reference IV

- Hochreiter, S. & Schmidhuber, J. Long short-term memory. Neural Comput. 9, 1735–1780 (1997).
  - This paper introduced LSTM recurrent networks, which have become a crucial ingredient in recent advances with recurrent networks because they are good at learning long-range dependencies.

Thank you!